An introduction to gEcon

Grzegorz Klima, Karol Podemski, Kaja Retkiewicz-Wijtiwiak

Warsaw, October 27, 2016

Outline

1. DSGE models
2. CGE models
3. gEcon project objectives
4. Main characteristics
5. gEcon language
6. Model solution procedure
7. Model calibration using gEcon.iosam package
8. Model estimation using gEcon.estimation package
9. gEcon use cases

DSGE models (1/2)

- DSGE (Dynamic Stochastic General Equilibrium) models are dynamic macroeconomic models derived from microeconomic principles (optimisation, market clearing, rational expectations).
- (Very) short history of DSGE modelling:
  - '70s — the appearance of new-classical macroeconomic school — first dynamic rational expectations (RE) and dynamic general equilibrium models
  - '80s — the seminal work of Kydland and Prescott (1982) (dawn of RBC school)
  - '90s — monopolistic competition and price rigidities are added to RBC models by Rotemberg and Woodford (1997)
  - '00s — Christiano, Eichenbaum, Evans (2005) model and estimated modification by Smets and Wouters (2003)
  - Nowadays, DSGE models are used across the globe by central banks and governments for policy analysis

DSGE models (2/2)

- DSGE models are complex and non-linear — they have to be solved using numerical methods
- In the '80 the numerical approach usually involved the linear-quadratic (LQ) approximation and the entire model model had to be implemented from scratch (mostly in FORTRAN)
- In the '90 MATLAB gained popularity and the first DSGE toolbox was released by Harald Uhlig, followed by a toolbox by Christopher Sims — both are based on the 1st order perturbation (the non-linear model is approximated by a linear RE model), the researcher had to derive the First Order Conditions (FOCs) and perturbation matrices
- Dynare project was a natural next step in the development of DSGE tools — only FOCs have to be derived, perturbation matrices are determined using symbolic computations

CGE models

- CGE (Computable General Equilibrium) models are a class of (generally static) applied economic models descending from the Input-Output models but based on microeconomic principles (optimisation and market clearing)
- CGE models have a long history with first models build in the '60s (Leif Johansen (1960)) and '70 (Taylor and Black (1974))
- CGE models are mostly used for comparative-static analyses of impact of external shocks or policy changes
- Initially, CGE models were written and implemented on a computer from scratch
- Curretly, most models are implemented in GAMS or GEMPACK frameworks, which allow for compact expression of similar equations differing by indices (of producers, households) and parameter values only

gEcon project objectives

- The ultimate goal of gEcon is to **reduce the development time of large-scale DSGE and CGE models for policy analysis and provide a unified framework for development of these two classes of models**
- The models are to be written — whenever possible — in terms of optimisation problems of agents without the need to manually derive the FOCs; symbolic manipulations on the part of the user should be kept to bare minimum
- The models should be easily scalable — the size of the model should not grow with the no. of sectors / types of households — a template mechanism is essential
- The process of solving the model should be interactive — the user should be able to change some parameters without the need to recompute the model
- The design of the package should allow for adding new functionalities and building solutions / packages on top of it

Main characteristics

- gEcon was developed as an R package — this choice (most DSGE packages are written in MATLAB) was motivated by R language flexibility and natural synergies between economic modelling and econometric work
- gEcon is based on a comprehensive symbolic computations library supporting symbolic differentiation, FOCs derivation, equation templates (including template differentiation), and symbolic reduction
- R interface is object-based, built around gecon_model class with a comprehensive set of functions useful for model analysis and debugging
- gEcon if focused on model equations derivation and solution but can be easily extended — as opposed to "black box" solutions — e.g. to allow for model estimation (gEcon.estimation package) and calibration using Input-Output or Social Accounting Matrices (gEcon.iosam package)

gEcon language (1/2)

- gEcon language should be easily understood by anyone with some exposure to R or MATLAB, variables (K[] — K_t) and parameters (alpha — α) are written in a natural way
- All standard mathematical operations (+, -, *, /, ^) and functions (log, exp, sin, ...) are supported
- Models are organised in blocks corresponding to optimising agents or equilibrium conditions:

```
block FIRM
{
    controls
    {
        K_d[], L_d[], Y[];
    };
    objective
    {
        pi[] = Y[] - L_d[] * W[] - r[] * K_d[];
    };
    constraints
    {
        Y[] = Z[] * K_d[] ^ alpha * L_d[] ^ (1 - alpha);
    };
};
```

gEcon language (2/2)

- Template support in gEcon is natural:

```
alpha<s>         # parameter alpha indexed with free index s
alpha<'AGR'>     # parameter alpha indexed with 'AGR'
Y<c>[]           # variable Y (at time 0) indexed with free index c
Y<'PL'>[]        # variable Y (at time 0) indexed with 'PL'
EX<'PL',c>[]     # variable EX (at time 0) indexed with index 'PL' and free index c
eta<'PL','DE'>   # parameter eta indexed with index 'PL' and index 'DE'
```

- . . . and allows for compact formulation of model equations:

```
CD[] = PROD<f::FACTORS>(C<f>[] ^ alpha<f>);
CES[] = (SUM<g::GOODS>(share<g> * D<g>[] ^ ((eta - 1) / eta))) ^ (eta / (eta - 1));
```

gEcon will understand these expression as:

$$CD_t = \prod_{f \in FACTORS} {C_t^{\langle f \rangle}}^{\alpha^{\langle f \rangle}},$$

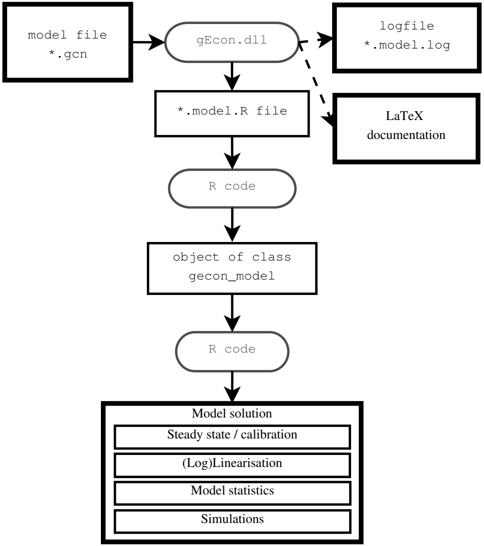
$$CES_t = \left(\sum_{g \in GOODS} share^{\langle g \rangle} {D_t^{\langle g \rangle}}^{(\eta - 1)/\eta} \right)^{\eta/(\eta-1)}.$$

Model solution procedure (1/3)

1. Models (optimisation problems of agents, equilibrium relationships, etc.) are written in gEcon language in a .gcn file
2. Models are read from R using the make_model function:

   ```
   > model <- make_model("PATH_TO_FILE/model.gcn")
   ```
3. the make_model function calls a shared library (DLL) that performs symbolic computations and then creates objects of class gecon_model in our workspace in R; in addition, logfile and LaTeXdocumentation of the model can be produced by the DLL
4. R scripts can then be used for solving the models (steady state / equilibrium computation, perturbation), simulation, and analysis

Model solution procedure (2/3)



Model solution procedure (3/3)

gEcon R interface was meant to be simple and intuitivee. Model construction and solution comes in few simple steps:

- Read model from .gcn file:

```
> model <- make_model("model.gcn")
```

- Find the steady state:

```
> model <- steady_state(model)
```

- Solve the 1st order perturbation:

```
> model <- solve_pert(model)
```

- Compute model statistics:

```
> model <- compute_model_stats(model = model,
                                ref_var = 'Y')
```

- Get information about selected model variables:

```
> var_info(model, c("Y", "C", "I"))
```

Model calibration using gEcon.iosam package

- Calibration of large-scale CGE and DSGE models using Input-Output Tables and Social Accounting Matrices can become a difficult and tedious task due to the number of parameters involved
- gEcon.iosam package is designed to assist users in this task by providing iosam class for representing Input-Output Tables and Social Accounting Matrices and a set of functions for importing and manipulating them
- To streamline the process of calibration of CGE (and multisector DSGE) models written using gEcon template mechanism, the function get_flow_values is provided

```
> model <- set_free_par(model, c(k_f_data    = sam["Firms", "K"], ks_data     = sam["SUM", "K"],
                                 ls_data     = sam["SUM", "L"], omega       = 2,
                                 get_flow_values(sam["L", sam_prod], "l_data", gcn_prod),
                                 get_flow_values(sam["SUM", sam_prod], "y_data", gcn_prod),
                                 get_flow_values(sam[sam_prod, sam_prod], "x_data", gcn_prod,  gcn_prod),
                                 get_flow_values(sam["Large_hh", "Firms"], "cap_data", "1"),
                                 get_flow_values(sam["Large_hh", "L"], "l_data", "1"),
                                 get_flow_values(scale_h, "scale",  gcn_hhds),
                                 get_flow_values(sam[sam_hhds, "K"], "k_data", gcn_hhds),
                                 get_flow_values(sam[7:8, sam_hhds], "d_data", c("B", "C"), gcn_hhds)
                                 ))
```

Model estimation using gEcon.estimation package (1/2)

- The gEcon.estimation package uses the state-space representation of models for likelihood computation (using Kalman filter) and estimation (Bayesian estimation or maximum likelihood approach).
- Additional functionalities include: forecasting functions, Kalman smoother for the model variables, historical shock decomposition.
- To estimate a model the user has to supply data (as ts objects), solved DSGE model (an object of gecon_model class), and prior distribution for parameters (six families of distributions are provided).

Model estimation using `gEcon.estimation` package (2/2)

- The Bayesian estimation is implemented in a standard way:
  - solver finds the mode and the standard deviation of the posterior kernel,
  - random walk Metropolis-Hastings routines are run to simulate the posterior.
- The results are stored in an `R` object.
- The flexible design makes the analysis of estimated model properties convenient (e.g. there is no need to re-estimate the model when changing forecast or smoother settings).

gEcon use cases

- Implementation of the Smets-Wouters '03 model — gEcon implementation led to revealing two mistakes in the manual derivation of model equations, https://ideas.repec.org/p/pra/mprapa/64440.html
- International trade CGE model calibrated using the GTAP database — implemented at the Department for Strategic Analyses at the Chancellery of the Prime Minister of the Republic of Poland
- CGE model for fiscal policy impact assessment — implemented at the Department for Strategic Analyses at the Chancellery of the Prime Minister of the Republic of Poland